Online Ad Assignment with an Ad Exchange

Wolfgang Dvořák and Monika Henzinger

University of Vienna, Faculty of Computer Science, Vienna, Austria

April 20, 2016

Abstract

Ad exchanges are becoming an increasingly popular way to sell advertisement slots on the internet. An ad exchange is basically a spot market for ad impressions. A publisher who has already signed contracts reserving advertisement impressions on his pages can choose between assigning a new ad impression for a new page view to a contracted advertiser or to sell it at an ad exchange. This leads to an online revenue maximization problem for the publisher. Given a new impression to sell decide whether (a) to assign it to a contracted advertiser and if so to which one or (b) to sell it at the ad exchange and if so at which reserve price. We make no assumptions about the distribution of the advertiser valuations that participate in the ad exchange and show that there exists a simple primal-dual based online algorithm, whose lower bound for the revenue converges to $R_{ADX} + R_A(1 - 1/e)$, where R_{ADX} is the revenue that the optimum algorithm achieves from the ad exchange and R_A is the revenue that the optimum algorithm achieves from the contracted advertisers.

1 Introduction

The market for display ads on the internet is worth billions of dollars and continues to rise. Not surprisingly, there are multiple ways of selling display advertisements. Traditionally, publishers signed long-term contracts with their advertisers, fixing the number of *impressions*, i.e. assigned ad slots views, as well as their price. In the last few years, however, spot markets, so called *Ad Exchanges* [1], have been developed, with Amazon, Ebay, and Yahoo (to just name a few) all offering their own ad exchange. Thus, every time a user requests to download a page from a publisher, the publisher

needs to decide (a) which of the ad impressions on this page should be assigned to which contracted advertiser, and (b) which should be sold at the ad exchange and at which *reserve price*[1].

Ad exchanges are interesting for publishers as (1) basically an unlimited number of ad impressions can be sold at ad exchanges, and (2) if the publishers have additional information about the user, they might sell an impression at a much higher price at the ad exchange than they could receive from their contracted advertisers. As ad impressions that did not receive a bid at or above the reserve price at the ad exchange can still be assigned to contracted advertisers, a revenue-maximizing publisher can offer *every* ad impression first at an ad exchange at a "high enough" reserve price and then afterwards assign the still unsold impressions to contracted advertisers. The question for the advertiser becomes, thus, (i) what reserve price to choose, and (ii) to which advertisers to assign the unsold impressions. We model this setting as an online problem and achieve the following two results: If the revenue achievable by the ad exchange for each ad impression is known, we give a constant competitive algorithm. Then we show how to convert this algorithm into a second algorithm that works in the setting where the revenue achievable from the ad exchange is not known. Assume that the auction executed at the ad exchange fulfills the following property P: *If an ad impression is sold at the ad exchange, then the revenue achieved is independent of the reserve price chosen by the publisher.* Thus, the reserve price influences only *whether* the ad impression is sold, *not* the price that is achieved. For example, a first price auction with reserve prices fulfills this condition. If the auction at the ad exchange fulfills this condition, then our second algorithm is constant competitive when compared with the optimum offline algorithm.

When modeling contracted advertisers we use the *model with free disposal* introduced in [2]: Each advertiser a comes with a number n_a and the revenue that an algorithm receives from a consists of the n_a *most valuable* ad impressions assigned to a. Additional impressions assigned to a do not generate any revenue.

More formally we define the following *Online Ad Assignment Problem with Free Disposal and an Ad Exchange.* There is a set of contracted advertisers A and an ad exchange α. Each advertiser a comes with a number n_a of ad impressions such that a pays only for the n_a *most valuable* ad impressions assigned to a, or for all assigned ad impressions if fewer than n_a are

[1] The reserve price is the minimum required price at which an impression is sold at an ad auction. If no offer is at or above the reserve price, the impression is not sold.

assigned to a. To simplify the notation we set $n_\alpha = \infty$. Now a finite sequence $\mathcal{S} = S_0, S_1, \ldots$ of sets S_l with $l = 0, 1, \ldots,$ of ad impressions arrives in order. When S_l arrives, the weights $w_{i,a}$ for each $i \in S_l$ and $a \in A \cup \{\alpha\}$ are revealed and the online algorithm has to assign each $i \in S_l$ *before* further sets S_{l+1}, S_{l+2}, etc. arrive. Let $\mathcal{A} : I \to A \cup \{\alpha\}$ be an *assignment* of impressions to advertisers. An assignment is *valid* if no two impressions in the same set S_l are assigned to the same advertiser $a \in A$. Let $I_{\mathcal{A}}(a)$ be the set of the n_a highest weighted impressions assigned to advertiser a by $\mathcal{A}$. Then the *revenue* $R(\mathcal{A})$ of $\mathcal{A}$ is $\sum_{a \in A \cup \{\alpha\}} \sum_{i \in I_{\mathcal{A}}(a)} w_{i,a}$. The goal of the algorithm is to produce a valid assignment $\mathcal{A}$ with maximum revenue $R(\mathcal{A})$. The *competitive ratio* of an online algorithm is the minimum over all sequences $\mathcal{S}$ of the ratio of the revenue achieved by the online algorithm on $\mathcal{S}$ and the revenue achieved by the optimal offline algorithm on $\mathcal{S}$, where the latter algorithm is given all of $\mathcal{S}$ *before* it makes the first decision.

Feldman et al. [2] studied a special case of our problem, namely the setting *without an ad exchange* and where each set S_l has size one, i.e. where the impressions arrive consecutively. For that setting they gave a primal-dual based 0.5 competitive algorithm whose competitive ratio converges to $(1-1/e)$ ratio when *all* the n_a values go to infinity. More precisely let $n_A = \min_{a \in A} n_a$. Then their algorithm is $1 - (\frac{n_A}{n_A+1})^{n_A}$-competitive. They also showed that this ratio is tight when considering deterministic algorithms [2]. Let R_a for an advertiser $a \in A \cup \{\alpha\}$ be the revenue that the optimal algorithm receives from a. We extend their results in several ways. (1) We consider a setting with one advertiser, called ad exchange, that has infinite capacity[2]. Moreover, we allow multiple ad slots on a page, with the condition that no two can be assigned to the same advertiser, i.e. for us $|S_l|$ can be larger than 1. (2) The revenue of our algorithm depends *directly* on the n_a value, not on n_A. More precisely, if no ad exchange exists, our algorithm receives a revenue of at least $\sum_a (1 - (\frac{n_a}{n_a+1})^{n_a}) R_a$. When an ad exchange is added, our algorithm achieves a revenue of at least $R_\alpha + \sum_a (1 - (\frac{n_a}{n_a+1})^{n_a}) R_a$. (3) We show how to modify our algorithm for the setting where $w_{i,\alpha}$ is unknown for all i. In this setting our algorithm computes a reserve price and sends *every* impression first to the ad exchange. The reserve price is set such that if the auction executed at the ad exchange fulfills property P then the above revenue bounds continue to hold, i.e. it achieves a revenue of at least $R_\alpha + \sum_a (1 - (\frac{n_a}{n_a+1})^{n_a}) R_a$.

[2] It is straightforward to extend the algorithm and its analysis to multiple ad exchanges.

Techniques Our algorithm is a modification of the standard primal-dual algorithms in [2] but it is itself not a standard primal-dual algorithms as it does not construct a feasible primal and dual solution to a *single* LP. Instead in the analysis we use several primal and dual LPs, one for each advertiser a and use the dual solutions to upper bound R_a. However, the corresponding primal feasible solution is *not* directly related to the revenue the algorithm achieves from a. Instead, the solution constructed by the algorithm is a feasible solution for a primal program that is the combination of all individual LPs. This property is strong enough to give the claimed bounds. The crucial new ideas in our algorithms are (i) the observation that when deciding to whom an ad slot is assigned the publisher should be biased towards advertiser with large n_a and in particular towards the ad exchange and (ii) that based on the structure of the algorithm it can be easily modified to compute an reserve price for the auction in the ad exchange if the $w_{i,\alpha}$ values are unknown.

Further Related Work We describe prior work on the question whether the publisher should assign an impression to a contracted advertiser or an ad exchange. In [3] a scenario is studied, where the $w_{i,a}$ follow a joint distribution and no disposal is allowed. Gosh et al. [4] assume that for each impression i the $w_{i,\alpha}$ values follow a known distribution and the contracted advertisers have a quality value depending on $w_{i,\alpha}$. They study the trade-off between the quality of the impressions assigned to the advertisers and revenue from the ad exchange. The work in [5], like our work, does not make Bayesian assumptions but studies online algorithms in the worst case setting. The main difference to our work is that there the contracted advertisers also arrive online and that there is no free disposal.

Finally, Devanur et al. [6] extend [2] to the scenario with multiple ad slots on a page and constraints on ads being assigned together, but they neither consider ad-exchanges nor consider the different capacities n_a in the competitive ratio.

Structure of the paper In Section 2 we discuss why the algorithm from [2] is not satisfying in our setting and present a simple online algorithm for the 1-slot case, which we improve in Section 3 to achieve a revenue of at least $R_\alpha + \sum_a (1 - (\frac{n_a}{n_a+1})^{n_a}) R_a$. In Section 4 we generalize this algorithm to the multi-slot setting. Finally, in Section 5 we show how to adapt it if the $w_{i,\alpha}$ values are unknown.

Algorithm 1

1. Initialize $\beta_a = 0$, $\beta_\alpha = 0$
2. When impression i arrives
   (a) Compute $j = \underset{a \in A \cup \{\alpha\}}{\operatorname{argmax}} \{w_{i,a} - \beta_a\}$.
   (b) if $j = \alpha$ then set $x_{i,\alpha} = 1$ and $z_i = w_{i,\alpha}$.
   (c) if $j \in A$ then set $x_{i,j} = 1$, $\forall\ i' \neq i:\ x_{i',j} = 0$, $z_i = w_{i,j} - \beta_j$ and $\beta_j = w_{i,j}$.

2 A Simple 1-Slot Online Algorithm

In Sections 2 and 3 we consider algorithms for the 1-slot setting, i.e., where each S_l just contains a single impression i. Given an instance of such an online ad assignment problem we can build an equivalent instance where all capacities $n_a = 1$. Simply replace each advertiser a by n_a copies $a_1, \dots a_{n_a}$ with the capacities 1 and for each impression i set $w_{i,a_p} = w_{i,a}$ for all $1 \leq p \leq n_a$. Thus in this section we assume $n_a = 1$ for each $a \in A$. Then we formulate the offline problem as an integer linear program (ILP), where the variable $x_{i,a}$ is set to 1 if i is assigned to advertiser a and to 0, otherwise.

$$\textbf{Primal:}\ \max \sum_{i,a \in A \cup \{\alpha\}} w_{i,a}\ x_{i,a}$$

$$\sum_{a \in A \cup \{\alpha\}} x_{i,a} \leq 1 \quad \forall i$$

$$\sum_i x_{i,a} \leq 1 \quad \forall a \in A$$

The first type of constraints ensures that each impression is assigned to at most one advertiser, while the second type of constraints ensures that each $a \in A$ is assigned at most one impression. It has the following dual LP.

$$\textbf{Dual:}\ \min \sum_i z_i + \sum_{a \in A} \beta_a$$

$$z_i + \beta_a \geq w_{i,a} \quad \forall i, \forall a \in A$$

$$z_i \geq w_{i,\alpha} \quad \forall i$$

For notational convenience we assume an additional variable β_α which remains 0 for the whole algorithm. We next consider a straight forward

generalization of the online algorithm in [2], called Algorithm 1, to our setting. This algorithm constructs a feasible integral solution for the Primal LP, corresponding to an ad assignment, and a feasible solution for the dual LP that is used to upper bound the revenue of the optimal assignment.

Algorithm 1 constructs feasible solutions for both the Primal and the Dual: when impression i is assigned to advertiser j then $x_{i,j}$ is set to 1, β_j is set to $w_{i,j}$, and z_i is set to $\max_{a \in A \cup \{\alpha\}}\{w_{i,a} - \beta_a\}$. Note that the loss in revenue of Algorithm 1 compared to the optimal assignment *exclusively* comes from the impression assigned to advertisers in A.

Proposition 1. *Let $\mathcal{A}$ be an ad assignment computed by Algorithm 1, then $R(OPT) \leq R_\alpha(\mathcal{A}) + 2 \cdot R_A(\mathcal{A})$.*

Proof. In the following we will use that $R(\mathcal{A})$ equals the value of the primal solution and that the value of the dual solution is an upper bound on $R(OPT)$. We prove the claim by induction on the assigned impressions. Clearly the base case where no impression is assigned is fine. Now consider an arbitrary i to be assigned and notice that, by (1.), β_a is such that $\beta_a = 0$ if no impression was assigned to a. Otherwise, by (2c), $\beta_a = w_{i',a}$ where i' is the highest weighted impression assigned to a. We simultaneously consider the increase of the primal and the dual solution when adding a new impression i.

If Algorithm 1 assigns i to an $a \in A$ this is by rule (c). Let β_a^n, β_a^o be the new and old value of β_a. The statement $x_{i,a} = 1, \forall\, i' \neq i : \; x_{i',a} = 0$ increases the revenue $R(\mathcal{A})$ by $w_{i,a} - \beta_a^o$, as β_a^o is the value $w_{i',a}$ of the impression i' we have to drop. On the other side the statement $z_i = w_{i,a} - \beta_a^o$ increases the objective of the Dual by the same amount. Additionally the objective of the Dual is affected by updating β_a with $\beta_a^n = w_{i,a}$. This additionally increases the objective of the Dual by $\beta_a^n - \beta_a^o = w_{i,a} - \beta_a^o$. Thus, if i is assigned to an $a \in A$, the increase of the objective of the Dual is twice the increase of the objective of the primal, the revenue $R(\mathcal{A})$.

If Algorithm 1 assigns i to α this is by rule (b). By $x_{i,\alpha} = 1$ we increase the revenue by $w_{i,\alpha}$ and by setting $z_i = w_{i,\alpha}$ we increase the objective of the Dual by the same amount. Hence, as in case (b) the β_a are not affected, we obtain that $R(OPT) \leq R_\alpha(\mathcal{A}) + 2 \cdot R_A(\mathcal{A})$. □

However, the Algorithm 1 does not guarantee that impressions are sent to ad exchange when the optimal algorithm does. Thus the optimal offline assignment might send many impressions to the ad exchange, while the online assignment of the above algorithm does not and thus might only be

Algorithm 2

1. Initialize $\beta_a = 0$ for all $a \in A \cup \{\alpha\}$
2. When impression i arrives
   (a) Compute $j = \underset{a \in A}{\operatorname{argmax}} \{w_{i,a} - \beta_a\}$.
   (b) if $(w_{i,j} - \beta_j) > 2 \cdot w_{i,\alpha}$ then assign i to j and set $\beta_j = w_{i,j}$.
   (c) if $(w_{i,j} - \beta_j) \leq 2 \cdot w_{i,\alpha}$ then assign i to α.

an 1/2 approximation (Proposition 1 guarantees that it is not worse.). Such a situation is given in Example 1.

Example 1. Consider $A = \{a\}$ with $n_a = 1$ and impressions $1 \leq i \leq n$ with $w_{i,\alpha} = 1 - \epsilon$ and $w_{i,a} = i$. Then the revenue $R(\mathcal{A})$ of Algorithm 1 after n impressions is n, while the optimal assignment achieves $n + (n-1)(1-\epsilon)$, where $(n-1)(1-\epsilon)$ is achieved by the ad exchange. For $\epsilon \to 0$ and $n \to \infty$ the ratio $R(\mathcal{A})/R(OPT)$ is 1/2 although half of the revenue in the optimal assignment OPT comes from the ad exchange. $\diamondsuit$

Thus the algorithm from [2] is only 1/2-competitive, even when an ad exchange, i.e., an advertiser with infinite capacity, is added.

Given an ad assignment $\mathcal{A}$ let $R_\alpha(\mathcal{A})$ denote the revenue the assignment gets from impressions assigned to the ad exchange and let $R_A(\mathcal{A})$ denote the revenue the assignment gets from impressions assigned to contracted advertisers. Thus we have $R(\mathcal{A}) = R_\alpha(\mathcal{A}) + R_A(\mathcal{A})$. Additionally, we use OPT to denote the optimal assignment. We present next Algorithm 2, an online algorithm that receives as revenue at least $R_\alpha(OPT) + (1/2)R_a(OPT)$, which is already an improvement over Algorithm 1. It is based on the observation that *assigning an impression that should be sent to the ad exchange to an advertiser in A is worse than sending an impression that should go to an advertiser in A to the ad exchange.* Thus, the algorithm is biased towards the ad exchange. Specifically the algorithm assigns an impression to an advertiser $a \in A$ only if it gives at least double the revenue on a than on α.

Theorem 1. *Let $\mathcal{A}$ be the ad assignment computed by Algorithm 2 then $R(\mathcal{A}) \geq R_\alpha(OPT) + 1/2 \cdot R_A(OPT)$.*

2.1 Proof of Theorem 1

Let I_{OPT}^{A}, resp. I_{OPT}^{α}, be the impressions assigned to A, resp. α, by the optimal (offline) assignment OPT. [3] We give an LP P_A for the advertisers A and impressions I_{OPT}^{A} and its dual D_A such that any feasible solution for D_A gives an upper bound d_A for $R_A(OPT)$.

Primal P_A**:** $\max \sum_{i\in I_{OPT}^{A}, a\in A} w_{i,a}\, x_{i,a}$

$$\sum_{a\in A} x_{i,a} \le 1 \quad \forall i \in I_{OPT}^{A}$$

$$\sum_{i\in I_{OPT}^{A}} x_{i,a} \le 1 \quad \forall a \in A$$

Dual D_A**:** $\min \sum_{i\in I_{OPT}^{A}} z_i + \sum_{a\in A} \beta_a$

$$z_i + \beta_a \ge w_{i,a} \quad \forall i \in I_{OPT}^{A}\ \forall a \in A$$

Note that the summation in P_A and the constraints in D_A are only over impressions in I_{OPT}^{A}. The objective value of the optimal solution of D_A, is an upper bound for the objective of P_A, and thus also for $R_A(OPT)$. However, there is no direct relationship between $R_A(\mathcal{A})$ and the objective of P_A for $\mathcal{A}$, as $\mathcal{A}$ might also assign impressions from I_{OPT}^{α} to A.

To upper bound $R_A(OPT)$ we construct a *feasible solution for* D_A. We do this in a iterative fashion, that is whenever Algorithm 2 assigns an impression $i \in I_{OPT}^{A}$ we update the feasible solution for D_A as follows:

(i) For the β_a variables we use the values currently set by the Algorithm 2;

(ii) For the variable z_i we set $z_i = w_{i,j} - \beta_j^o$, where β_a^o is the value of β_a before i is assigned.

As $w_{i,j} - \beta_j^o = max_{a\in A}\{w_{i,a} - \beta_a\}$, all the constraints for i are satisfied. Hence, doing this for all $i \in I_{OPT}^{A}$ gives a feasible solution for D_A and its objective d_A fulfills $d_A \ge R_A(OPT)$.

To show that the claim holds after each assignment of an impression i we investigate assigning one expression i and study the effect to both the upper bound and the revenue we achieve. To this end we introduce some notation:

- $\Delta d_A(i)$ is the increase of the objective d_A when the algorithm assigns impression i, i.e., the change in d_A caused by the change in the β-values and the assignment of the new z_i value (if $i \in I_{OPT}^{A}$).

[3] In case there are several optimal assignments we pick an arbitrary one.

- $\Delta d_\alpha(i) = w_{i,\alpha}$ if $i \in I^\alpha_{OPT}$ and $\Delta d_\alpha(i) = 0$ otherwise.
- $\Delta R(\mathcal{A}, i)$ is the increase of the revenue when assigning i.
- β^n_a, resp. β^o_a, to denote the value of β_a after, resp. before i is assigned

Note that by the definitions (a) $\sum_{i \in I} \Delta d_A(i) = d_A$, (b) $\sum_{i \in I} \Delta d_\alpha(i) = R_\alpha(OPT)$ and (c) $\sum_{i \in I} \Delta R(\mathcal{A}, i) = R(\mathcal{A})$. We will also exploit the fact that β_a is such that $\beta_a = 0$ if no impression was assigned to a and otherwise $\beta_a = w_{i',a}$, where i' is the impression currently assigned to a.

Next, to relate the increase of the upper bound with the gain of revenue, we distinguish whether one assigns an impression $i \in I^\alpha_{OPT}$ or an impression $i \in I^A_{OPT}$

Lemma 1. $\Delta R(\mathcal{A}, i) \geq \Delta d_\alpha(i) + 1/2 \cdot \Delta d_A(i)$*, for* $i \in I^A_{OPT}$.

Proof. As $i \in I^A_{OPT}$, by definition, we have $\Delta d_\alpha(i) = 0$. Thus, we actually have to show that $\Delta R(\mathcal{A}, i) \geq 1/2 \cdot \Delta d_A(i)$.

1. If Algorithm 2 assigns i to an $j \in A$ recall that we set $z_i = w_{i,j} - \beta^o_j$ and the algorithm sets $\beta^n_j = w_{i,j}$. Thus $\Delta d_A(i) = 2 \cdot (w_{i,j} - \beta^o_j)$ and $\Delta R(\mathcal{A}, i)$ is given by $w_{i,j}$ minus the value of the impression we have to drop (if any), given by β^0_a. As this values is stored in β^o_j we get $\Delta R(\mathcal{A}, i) = w_{i,j} - \beta^o_j$ and thus $\Delta R(\mathcal{A}, i) \geq 1/2 \cdot \Delta d_A(i)$.
2. If Algorithm 2 assigns i to α (although OPT does not), we know from Step 2c that $(w_{i,j} - \beta_j) \leq 2 w_{i,\alpha}$, where $j = \underset{a \in A}{\operatorname{argmax}} \{w_{i,a} - \beta_a\}$. As we set $z_i = w_{i,j} - \beta^o_j$ and the algorithm keeps all β_a unchanged we get $\Delta d_A(i) = w_{i,j} - \beta^o_j$ and as we assign i to α we have $\Delta R(\mathcal{A}, i) = w_{i,\alpha}$. Thus $\Delta R(\mathcal{A}, i) = w_{i,\alpha} \geq 1/2 \cdot (w_{i,j} - \beta_j) = 1/2 \cdot \Delta d_A(i)$.

Hence, $\Delta R(\mathcal{A}, i) \geq 1/2 \cdot \Delta d_A(i) = \Delta d_\alpha(i) + 1/2 \cdot \Delta d_A(i)$. □

Lemma 2. $\Delta R(\mathcal{A}, i) \geq \Delta d_\alpha(i) + 1/2 \cdot \Delta d_A(i)$*, for* $i \in I^\alpha_{OPT}$.

Proof. As $i \in I^\alpha_{OPT}$, by definition, $\Delta d_\alpha(i) = w_{i,\alpha}$. Recall that no z-value is affected in this case. We have to show that $\Delta R(\mathcal{A}, i) \geq w_{i,\alpha} + 1/2 \cdot \Delta d_A(i)$.

1. If Algorithm 2 assigns i to the ad exchange then the β_a are not changed. Thus $\Delta d_A(i) = 0$ and $\Delta R(\mathcal{A}, i)$ is simply $w_{i,\alpha}$. Hence, $\Delta R(\mathcal{A}, i) = w_{i,\alpha} \geq w_{i,\alpha} + 1/2 \cdot \Delta d_A(i)$.

2. If Algorithm 2 assigns i to an $a \in A$ we have $(w_{i,a} - \beta_a^o) > 2w_{i,\alpha}$ and the algorithm sets $\beta_a^n = w_{i,a}$. Thus $\Delta d_A(i) = w_{i,a} - \beta_a^o$. Furthermore, $\Delta R(\mathcal{A}, i)$ is given by $w_{i,a}$ minus the value of the impression we have to drop (if any), given by β_a^0. Thus $\Delta R(\mathcal{A}, i) = (w_{i,a} - \beta_a^o) = 1/2 \cdot (w_{i,a} - \beta_a^o) + 1/2 \cdot (w_{i,a} - \beta_a^o) \geq w_{i,\alpha} + 1/2 \cdot \Delta d_A(i)$.

Hence, $\Delta R(\mathcal{A}, i) \geq w_{i,\alpha} + 1/2 \cdot \Delta d_A(i) = \Delta d_\alpha(i) + 1/2 \cdot \Delta d_A(i)$. □

When combining Lemma 1 and Lemma 2 we obtain that for each $i \in I$:

$$\Delta R(\mathcal{A}, i) \geq \Delta d_\alpha(i) + 1/2 \cdot \Delta d_A(i).$$

Finally, when summing over all impression $i \in I$ and using the above inequality we obtain the claim.

$$R(\mathcal{A}) = \sum_{i \in I} \Delta R(\mathcal{A}, i) \geq \sum_{i \in I} \left(\Delta d_\alpha(i) + \frac{\Delta d_A(i)}{2} \right) \geq R_\alpha(OPT) + \frac{R_A(OPT)}{2}.$$

This completes the proof of Theorem 1.

3 An Online 1-Slot Algorithm Exploiting High Capacities

In this section we generalize the result from Section 2 to the setting where each advertiser $a \in A$ has an individual limit n_a for the number of ad impressions he is willing to pay for and we present Algorithm 3 that achieves an improvement in revenue for advertisers a with large n_a.

In Algorithm 3 we consider variables β_a which, for $a \in A$, are always set s.t.

$$\beta_a = \frac{1}{n_a(e_{n_a} - 1)} \sum_{j=1}^{n_a} w_j \left(1 + \frac{1}{n_a}\right)^{j-1} \tag{1}$$

where the w_j's are the weights of the impressions assigned to a in non-increasing order and $e_{n_a} = (1 + 1/n_a)^{n_a}$. That is, β_a stores a weighted mean of the n_a most valuable impressions assigned to a. Again we keep $\beta_\alpha = 0$ in the whole algorithm. Next we consider how assigning a new impression to a affects β_a.

Lemma 3. *Consider a new impression i being assigned to advertiser a. Let β_a^o, resp. β_a^n denote the value of β_a before, resp. after i was assigned and v the value of the impression dropped from β_a (0 if no impression is dropped), then*

$$\beta_a^n - \beta_a^o \leq \frac{\beta_a^o}{n_a} - \frac{v \cdot e_{n_a}}{n_a(e_{n_a} - 1)} + \frac{w_{i,a}}{n_a(e_{n_a} - 1)}.$$

Lemma 3 was already shown in [2] but to keep the paper self-explanatory we provide a proof.

Proof. Assume that $w_{i,a}$ is the impression with the k-th highest value.

$$\begin{aligned}\beta_a^n &= \frac{1}{n_a(e_{n_a} - 1)}\left[\sum_{j=1}^{k-1} w_j \left(1 + \frac{1}{n_a}\right)^{j-1} + w_{i,a}\left(1 + \frac{1}{n_a}\right)^{k-1} + \sum_{j=k}^{n_a-1} w_j \left(1 + \frac{1}{n_a}\right)^{j}\right] \\ &\leq \frac{1}{n_a(e_{n_a} - 1)}\left[w_{i,a} + \sum_{j=1}^{n_a-1} w_j \left(1 + \frac{1}{n_a}\right)^{j}\right]\end{aligned}$$

To obtain the inequality we exploited that $w_j > w_{i,a}$ for $j < k$ and that $(1 + 1/n_a)^{j-1} < (1 + 1/n_a)^{j'-1}$ for $j < j'$. We proceed with standard transformations.

$$\begin{aligned}\beta_a^n &\leq \frac{1}{n_a(e_{n_a} - 1)}\left[w_{i,a} + \sum_{j=1}^{n_a-1} w_j \left(1 + \frac{1}{n_a}\right)^{j}\right] \\ &= \frac{w_{i,a}}{n_a(e_{n_a} - 1)} + \frac{1}{n_a(e_{n_a} - 1)} \sum_{j=1}^{n_a-1} w_j \left(1 + \frac{1}{n_a}\right)^{j} \\ &= \frac{w_{i,a}}{n_a(e_{n_a} - 1)} + \left(1 + \frac{1}{n_a}\right)\beta_a^o - \frac{v \cdot e_{n_a}}{n_a(e_{n_a} - 1)}\end{aligned}$$

From the last statement we obtain $\beta_a^n - \beta_a^o \leq \frac{\beta_a^o}{n_a} - \frac{v \cdot e_{n_a}}{n_a(e_{n_a} - 1)} + \frac{w_{i,a}}{n_a(e_{n_a} - 1)}$. $\square$

Notice that in Algorithm 3 for each $a \in A$ we have that $1/2 \leq c_a < 1 - 1/e$, i.e. for $n_a = 1$ we have $c_a = 1/2$ and c_a grows with n_a and converges to $1 - 1/e$. The idea is to bias the algorithm towards advertisers with larger n_a and in particular towards the ad-exchange.

We use $R_a(\mathcal{A})$ for $a \in A \cup \{\alpha\}$ to denote the revenue the assignment $\mathcal{A}$ gets from advertiser a. Thus, $R(\mathcal{A}) = \sum_{a \in A \cup \{\alpha\}} R_a(\mathcal{A})$.

Theorem 2. *Let $\mathcal{A}$ be the assignment computed by Algorithm 3 then $R(\mathcal{A}) \geq \sum_{a \in A \cup \{\alpha\}} c_a \cdot R_a(OPT)$.*

Algorithm 3

1. Initialize $\beta_a = 0$ for all $a \in A \cup \{\alpha\}$
2. When impression i arrives
   (a) Compute $x = \underset{a \in A \cup \{\alpha\}}{\operatorname{argmax}} \{c_a \cdot (w_{i,a} - \beta_a)\}$
   (b) assign i to x and update β_x according to (1)

where weights c_a are defined as $c_a = \begin{cases} 1 - \frac{1}{e_{n_a}} & a \in A \\ 1 & a = \alpha \end{cases}$

Theorem 2 will be a direct consequence of Theorem 3, which we will prove in the next section.

Finally let us briefly discuss whether the constants c_a are chosen optimally. From a result in [7] on online algorithms for b-matchings it follows immediately that the constants c_a in Theorem 2 are optimal for deterministic algorithms. Moreover, in [8] it is shown that even randomized algorithms cannot achieve a better competitive ratio than $(1-1/e)$ [4]. So for large values of n_a even randomized algorithms cannot improve over Algorithm 3.

4 A Multi-Slot Online Algorithm

In practice publishers often have several ad slots at a single page and want to avoid to show multiple ads from the same advertiser on the same page to avoid annoying their users. This can be modeled as follows: A sequence $\mathcal{S} = S_0, S_1, \ldots$ of *sets* of impressions arrive in an online manner. Each set S has be assigned (a) before any future sets have arrived, and (b) such that non two impressions in S are assigned to the same advertiser in A. Note that we allow multiple impressions from S to be assigned to the ad exchange as we expect the ad exchange to return different advertisers for them. Let the set of all impressions $I = \sum_{S \in \mathcal{S}} S$. With Algorithm 4 we present an online algorithm for this setting with the same competitive ratio as Algorithm 3. Note, however, that, unlike Algorithm 3, it is compared to the optimal offline solution that respects the above restriction. More formally, we call a function $\mathbf{a} : S \to A \cup \{\alpha\}$ assigning impressions S to advertisers *valid*

[4] In [8] the authors study the Adwords problem but in [2] it is argued that the given example can be also be interpreted as Online Ad Assignment problem.

Algorithm 4

1. Initialize $\beta_a = 0$ for all $a \in A \cup \{\alpha\}$
2. When impressions $S = \{i_1, \ldots, i_l\}$ arrive
   (a) Compute $\mathbf{b} = \underset{\text{valid } \mathbf{a}}{\operatorname{argmax}} \left\{ \sum_{i \in S} c_{\mathbf{a}(i)} \cdot (w_{i,\mathbf{a}(i)} - \beta_{\mathbf{a}(i)}) \right\}$
   (b) assign each i to $\mathbf{b}(i)$ and, if $\mathbf{b}(i) \in A$, update $\beta_{\mathbf{b}(i)}$ according to (1).

where weights c_a are defined as $c_a = \begin{cases} 1 - \frac{1}{e_{n_a}} & a \in A \\ 1 & a = \alpha \end{cases}$

if there are no $i, i' \in S$, $i \neq i'$, $a \in A$ such that $\mathbf{a}(i) = \mathbf{a}(i') = a$. Our Algorithm 4 generates a valid assignment and is compared to the revenue of the *valid* assignment generated by the optimal offline algorithm. Notice that the computation of argmax in Algorithm 4 is a weighted bipartite matching problem and thus can be computed efficiently.

Recall that $R_a(OPT)$ for $a \in A \cup \{\alpha\}$ is the revenue that an optimal assignment generates from advertiser a. We show the following performance bound.

Theorem 3. *Let $\mathcal{A}$ be the assignment computed by Algorithm 4 and OPT the optimal multi-slot ad assignment, then $R(\mathcal{A}) \geq \sum_{a \in A \cup \{\alpha\}} c_a \cdot R_a(OPT)$.*

The proof of Theorem 3 generalizes ideas from the proof of Theorem 1. One of the main difference is that we now have to deal with several sets I^a_{OPT} instead of just one set I^A_{OPT} and thus also with several LPs.

4.1 Proof of Theorem 3

First we give a linear program P_a and its dual D_a for each $a \in A$ such that the final objective value of any feasible solution of D_a is an upper bound of $R_a(OPT)$. Note, that there is no direct relationship between the final objective values of the P_a's and the revenue of the algorithm. However, we are able to construct a feasible solution for each D_a with objective value d_a such that the revenue $R(\mathcal{A})$ of the algorithm is at least $\sum_{a \in A \cup \alpha} c_a \cdot d_a$. Together with the observation that $d_a \geq R_a(OPT)$ and a bound d_α on $R_\alpha(OPT)$ this proves the theorem.

Let I^a_{OPT} be the impressions assigned to $a \in A \cup \{\alpha\}$ by the optimal (offline) assignment OPT. We consider the following LPs for each $a \in A$.

Primal P_a**:** $\max \sum_{i \in I^a_{OPT}} w_{i,a}\, x_{i,a}$

$$x_{i,a} \leq 1 \quad \forall i \in I^a_{OPT}$$

$$\sum_{i \in I^a_{OPT}} x_{i,a} \leq n_a$$

Dual D_a**:** $\min \sum_{i \in I^a_{OPT}} z_i + n_a \beta_a$

$$z_i + \beta_a \geq w_{i,a} \quad \forall i \in I^a_{OPT}$$

Note that the summation in the primal and the constraints in the Dual are only over the impressions *in* I^a_{OPT}, i.e., the impressions assigned to a by OPT. The objective value of the optimal solution for D_a is an upper bound for the objective of P_a, and thus also for $R_a(OPT)$. This implies that any feasible solution of D_a, also the one we construct next, gives an upper bound for $R_a(OPT)$. As there might be impressions assigned to a by the algorithm that do *not* belong to I^a_{OPT}, the objective value of P_a is, however, not necessarily related to $R_a(\mathcal{A})$.

Next, we give a *feasible solution for* D_a for all $a \in A$, as follows.

(i) For the β_a variables we use the values currently set by the Algorithm 4;

(ii) Let $\mathbf{a}$ be the assignment of the impressions in S by the optimal solution. For each $i \in I$, we set $z_i = w_{i,\mathbf{a}(i)} - \beta_{\mathbf{a}(i)}$ exactly when the algorithm assigns i.

Note that this results in a feasible dual solution for *all* a as each i belongs to exactly one set $I^{\mathbf{a}(i)}_{OPT}$ and z_i is chosen exactly so as to make the solution of $D_{\mathbf{a}(i)}$ feasible, together with the current $\beta_{\mathbf{a}(i)}$ values. As $\beta_{\mathbf{a}(i)}$ only increases in the course of the algorithm the solution remains feasible at the end of the algorithm. Let d_a be the value of this feasible solution for D_a for some $a \in A$ then we have $d_a \geq R_a(OPT)$.

To show that the claim holds after each assignment of a set of impressions S we investigate assigning one such set and study the effect to both the upper bound and the revenue we achieve. To this end we introduce some notation:

- $\Delta d_a(S)$ be the increase of the objective value d_a when the algorithms assigns S, i.e., the change in d_a caused by the change in the β_a-values *and* the assignment of the z_i-values for all $i \in S \cap I^A_{OPT}$.
- $\Delta d_\alpha(S) = \sum_{i \in S \cap I^\alpha_{OPT}} w_{i,\alpha}$.
- $\Delta R(\mathcal{A}, S)$ is the increase of revenue when S is assigned.

Note that by the definitions (a) $\sum_{S\in\mathcal{S}} \Delta d_a(S) = d_a$, (b) $\sum_{S\in\mathcal{S}} \Delta d_\alpha(S) = d_\alpha$ and (c) $\sum_{S\in\mathcal{S}} \Delta R(\mathcal{A}, S) = R(\mathcal{A})$.

Lemma 4. *$\Delta R(\mathcal{A}, S) \geq \sum_{a\in A\cup\alpha} c_a \cdot \Delta d_a(S)$, for all $S \in \mathcal{S}$.*

Proof. To simplify the notation let $\Delta d(S) = \sum_{a\in A\cup\alpha} c_a \cdot \Delta d_a(S)$.

First consider $\Delta R(\mathcal{A}, S)$: For $a \in A$ let v_a be the value of the n_a-th valuable impression assigned to a (the impression we would "drop" by assigning a new one), and let $v_\alpha = 0$. If i is assigned to α then the gain in revenue is $w_{i,\mathbf{b}(i)}$ which equals $w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)}$. If i is assigned to $a \in A$ then the gain in revenue is the difference between the revenue of the new impression and the impression we have to drop, i.e., again $w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)}$. Thus for S altogether it holds

$$\Delta R(\mathcal{A}, S) = \sum_{i\in S} (w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)})$$

Now consider $\Delta d(S)$*:* Recall that $\mathbf{a}$ is the assignment of the optimal solution for the impressions S and let $\mathbf{b}$ be the assignment from Algorithm 4. For all $a \in A$ let β_a^o, β_a^n denote the value of β_a right *before*, resp. right *after* this assignment. Recall that for $a = \alpha$, it holds that $\beta_a = 0$ throughout the algorithm. Now note that

$$\Delta d(S) = \sum_{i\in S} \left(c_{\mathbf{a}(i)} \cdot (w_{i,\mathbf{a}(i)} - \beta^o_{\mathbf{a}(i)}) + c_{\mathbf{b}(i)} \cdot n_{\mathbf{b}(i)} \cdot (\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) \right),$$

where the first term comes from the new variables z_i which we set to $(w_{i,\mathbf{a}(i)} - \beta^o_{\mathbf{a}(i)})$, and the second term comes from the updates of β_a. By the choice of $\mathbf{b}$ in the algorithm we get

$$\begin{aligned}
\Delta d(S) &\leq \sum_{i\in S} \left(c_{\mathbf{b}(i)} \cdot (w_{i,\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) + c_{\mathbf{b}(i)} \cdot n_{\mathbf{b}(i)} \cdot (\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) \right) \\
&= \sum_{i\in S} c_{\mathbf{b}(i)} \cdot \left((w_{i,\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) + n_{\mathbf{b}(i)} \cdot (\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) \right).
\end{aligned}$$

Next we bound the contribution of each $i \in S$ separately by analyzing two cases:

- If $\mathbf{b}(i) = \alpha$ then we know that $\beta^o_{\mathbf{b}(i)} = \beta^n_{\mathbf{b}(i)} = v_{\mathbf{b}(i)} = 0$ and $c_{\mathbf{b}(i)} = 1$. Thus

  $$c_{\mathbf{b}(i)} \cdot \left((w_{i,\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) + c_{\mathbf{b}(i)} \cdot n_{\mathbf{b}(i)} \cdot (\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) \right) = (w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)}).$$

- If $\mathbf{b}(i) \in A$ then we can apply Lemma 3 to bound $(\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)})$ as follows

$$c_{\mathbf{b}(i)} \cdot \Big((w_{i,\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) + n_{\mathbf{b}(i)} \cdot (\beta^n_{\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) \Big) \leq$$
$$c_{\mathbf{b}(i)} \cdot \left((w_{i,\mathbf{b}(i)} - \beta^o_{\mathbf{b}(i)}) + \beta^o_{\mathbf{b}(i)} - \frac{v_{\mathbf{b}(i)} \cdot e_{n_{\mathbf{b}(i)}}}{e_{n_{\mathbf{b}(i)}} - 1} + \frac{w_{i,\mathbf{b}(i)}}{e_{n_{\mathbf{b}(i)}} - 1} \right) =$$
$$c_{\mathbf{b}(i)} \cdot \left(\frac{w_{i,\mathbf{b}(i)} \cdot e_{n_{\mathbf{b}(i)}}}{e_{n_{\mathbf{b}(i)}} - 1} - \frac{v_{\mathbf{b}(i)} \cdot e_{n_{\mathbf{b}(i)}}}{e_{n_{\mathbf{b}(i)}} - 1} \right) = (w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)})$$

In the last step we used that by definition $c_a = 1 - 1/e_{n_a}$ for $a \in A$. By the above we obtain

$$\Delta d(S) \leq \sum_{i \in S} (w_{i,\mathbf{b}(i)} - v_{\mathbf{b}(i)}) = \Delta R(\mathcal{A}, S). \qquad \square$$

Now consider that the set of impression is given by a series $(S_j)_{0 \leq j \leq n}$ of pairwise disjoint sets of impressions that show up simultaneously. Exploiting Lemma 4 we get:

$$R(\mathcal{A}) = \sum_{j=0}^{n} \Delta R(\mathcal{A}, S_j) \geq \sum_{j=0}^{n} \Delta d(S_j) =$$
$$\sum_{j=0}^{n} \sum_{a \in A \cup \{\alpha\}} c_a \cdot \Delta d_a(S_j) \geq \sum_{a \in A \cup \{\alpha\}} c_a \cdot R_a(OPT)$$

This completes the proof of Theorem 3.

5 An Algorithm for Computing Reserve Prices

In our model we assumed the publisher knows exactly how much revenue he can get from the ad exchange, i.e., the $w_{i,\alpha}$ values are given for all $i \in I$. The critical reader may interpose that this is not the fact in the real world or in the ad exchange model proposed in [1]. Instead whenever sending an impression to the ad exchange an auction is run. However, the publisher can set a reserve price and if all the bids are below the reserve price then he can still assign it to one of the contracted advertisers.

One nice property of Algorithms 2 & 3 is that they allow to compute the minimal price we have to extract from the ad exchange such that it is better to assign an impression to the ad exchange than to a contracted advertiser.

This price is given by $\max_{a \in A} \{c_a \cdot (w_{i,a} - \beta_a)\}$. It follows that this price is also a natural choice for the reserve price. Assume the auction executed at the ad exchange fulfills the following *property (P): If an ad impression is sold at the ad exchange, then the revenue achieved is independent of the reserve price chosen by the publisher.* Thus, the reserve price influences only whether the ad impression is sold, not the price that is achieved. Then Theorem 3 applies, i.e., the revenue of the algorithm is at least $\sum_{a \in A \cup \{\alpha\}} c_a \cdot R_a(OPT)$, even though the algorithm is not given the $w_{i,\alpha}$ values and it is compared to an optimal algorithm that does. The reason is that the algorithm makes exactly the same decisions and receives exactly the same revenue as Algorithm 3 that is given the $w_{i,\alpha}$ values.

Theorem 4. *Let $\mathcal{A}$ be the assignment computed by the Algorithm described above, i.e., without knowledge of the $w_{i,\alpha}$ values. If the auction at the ad exchange fulfills property P, then $R(\mathcal{A}) \geq \sum_{a \in A \cup \{\alpha\}} c_a \cdot R_a(OPT)$.*

However, this technique does not work for Algorithm 4. When a set of impressions S is assigned the right reserve price for an impression $i \in S$ depends on the assignment of the other impressions in S. In particular the optimal reserve price for one impression might depend on the outcome of an auction for another item in the same set. We leave integrating reserve prices in the multi-slot setting as an open question for future research.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 340506 and from the Vienna Science and Technology Fund (WWTF) through project ICT10-002.

The authors are grateful to Claire Kenyon and Moses Charikar for useful discussions on formulating the model.

A preliminary version of this paper has appeared in the proceedings of the 12th International Workshop on Approximation and Online Algorithms [9].

References

[1] S. Muthukrishnan, Ad exchanges: Research issues, in: S. Leonardi (Ed.), Internet and Network Economics, 5th International Workshop,

WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings, Vol. 5929 of Lecture Notes in Computer Science, Springer, 2009, pp. 1–12. doi:10.1007/978-3-642-10841-9_1.

[2] J. Feldman, N. Korula, V. S. Mirrokni, S. Muthukrishnan, M. Pál, Online ad assignment with free disposal, in: S. Leonardi (Ed.), Internet and Network Economics, 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings, Vol. 5929 of Lecture Notes in Computer Science, Springer, 2009, pp. 374–385. doi:10.1007/978-3-642-10841-9_34.

[3] S. R. Balseiro, J. Feldman, V. S. Mirrokni, S. Muthukrishnan, Yield optimization of display advertising with ad exchange, Management Science 60 (12) (2014) 2886–2907. doi:10.1287/mnsc.2014.2017.

[4] A. Ghosh, P. Mcafee, K. Papineni, S. Vassilvitskii, Bidding for representative allocations for display advertising, in: Proceedings of the 5th International Workshop on Internet and Network Economics, WINE '09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 208–219. doi:10.1007/978-3-642-10841-9_20.

[5] S. Alaei, E. Arcaute, S. Khuller, W. Ma, A. Malekian, J. Tomlin, Online allocation of display advertisements subject to advanced sales contracts, in: Proceedings of the Third International Workshop on Data Mining and Audience Intelligence for Advertising, ADKDD '09, ACM, New York, NY, USA, 2009, pp. 69–77. doi:10.1145/1592748.1592758.

[6] N. R. Devanur, Z. Huang, N. Korula, V. S. Mirrokni, Q. Yan, Whole-page optimization and submodular welfare maximization with online bidders, in: M. Kearns, R. P. McAfee, É. Tardos (Eds.), ACM Conference on Electronic Commerce, EC '13, Philadelphia, PA, USA, June 16-20, 2013, ACM, 2013, pp. 305–322. doi:10.1145/2482540.2482603.

[7] B. Kalyanasundaram, K. Pruhs, An optimal deterministic algorithm for online b-matching, in: V. Chandru, V. Vinay (Eds.), Foundations of Software Technology and Theoretical Computer Science, 16th Conference, Hyderabad, India, December 18-20, 1996, Proceedings, Vol. 1180 of Lecture Notes in Computer Science, Springer, 1996, pp. 193–199. doi:10.1016/S0304-3975(99)00140-1.

[8] A. Mehta, A. Saberi, U. V. Vazirani, V. V. Vazirani, Adwords and generalized online matching, J. ACM 54 (5). doi:10.1145/1284320.1284321.

[9] W. Dvořák, M. Henzinger, Online ad assignment with an ad exchange, in: E. Bampis, O. Svensson (Eds.), Approximation and Online Algorithms - 12th International Workshop, WAOA 2014, Wrocław, Poland, September 11-12, 2014, Revised Selected Papers, Vol. 8952 of Lecture Notes in Computer Science, Springer, 2015, pp. 156–167. doi:10.1007/978-3-319-18263-6_14.